Journal of Data Science,17(3). P. 575 - 592,2019
DOI:10.6339/JDS.201907_17(3).0007

MODELING ON GENERALIZED EXTENDED INVERSE WEIBULL SOFTWARE RELIABILITY GROWTH MODEL

David D. Hanagal and Nileema N. Bhalerao

Department of Statistics, Savitribai Phule Pune University, Pune-411007, India.
Email: david_hanagal@yahoo.co.in; bhaleraonnb@gmail.com

ABSTRACT

In this paper we introduce the generalized extended inverse Weibull finite failure software reliability growth model which includes both increasing/decreasing nature of the hazard function. The increasing/decreasing behavior of failure occurrence rate fault is taken into account by the hazard of the generalized extended inverse Weibull distribution. We proposed a finite failure non-homogeneous Poisson process (NHPP) software reliability growth model and obtain unknown model parameters using the maximum likelihood method for interval domain data. Illustrations have been given to estimate the parameters using standard data sets taken from actual software projects. A goodness of fit test is performed to check statistically whether the fitted model provides a good fit with the observed data. We discuss the goodness of fit test based on the Kolmogorov-Smirnov (K-S) test statistic. The proposed model is compared with some of the standard existing models through error sum of squares, mean sum of squares, predictive ratio risk and Akaikes information criteria using three different data sets. We show that the observed data fits the proposed software reliability growth model. We also show that the proposed model performs satisfactory better than the existing finite failure category models.

Keywords: Akaikes information criterion; Generalized extended inverse Weibull distribution; Hazard function; Predictive ratio risk.

1. Introduction

Due to the rapid development of computer and information technology, society increasingly depends on software-intensive systems. Software is embedded in many modern systems, including expensive scientific computing systems, financial banking systems, industrial applications, university computer centers and home personal computers. Since the demands for complex and large scale software systems are increasing more rapidly, the possibility of programmers' design error in the system will grow appreciably. Consequently, the possibility of crises due to software failure will continue to increase. These failures can generate enormous losses of revenues to many enterprises. A metric is needed to enhance software quality. One such quantitative metric of quality that is commonly used in software engineering practice is software reliability.

Software reliability is a probabilistic measure and can be defined as the probability that the software faults do not cause a failure during a specified exposure period in a specified use environment. For more precise definition one can refer Goel (1985). Research activities in software reliability engineering have been conducted over the past 30 years, and many models have been developed for the estimation of software reliability [see; Wood (1996),

Pham (2006), and Chen (2010)].

Hanagal and Bhalerao (2016) proposed generalized inverse Weibull software reliability growth model (SRGM). Hanagal and Bhalerao (2017, 2018) recently proposed extended inverse Weibull SRGM and delayed S-shaped SRGM with time dependent fault content rate function.

Traditionally there are two common types of failures data: time domain data and interval domain data. These data are usually used by practitioners when analyzing and predicting reliability applications. This paper presents the work using interval-domain data. The interval domain approach is characterized by counting the number of failures occurring during a fixed period (e.g., test session, hour, week, and day). Using this method, the collected data are a count of the number of failure in the interval. Most of the existing finite failure category models describe the failure rate either a constant, increasing or decreasing over time, for further information regarding this, one can refer Lyu (1996) and Pham (2006). In reality this may not be true because in early stage of testing, the testers are new to the software and they need time to adjust. It implies less testing of failures at the beginning of testing stage. As testing progress testers get good exposure to software resulting in increase in detection of failures, thus detection of faults increasers in this period of time. When most of the faults are eliminated and failures of software decreases as time increases.

This phenomenon of increasing/decreasing failure behaviors is not discussed much in the literature in the existing finite failure models. This motivated us for the development of a new model taking into account the increasing/decreasing behaviors of the software failures by its hazard rate function, see Hjorth (1980). The notion of failure rate is crucial in reliability and survival analysis. However, obtaining the failure rate in many practical situations is often not so simple because of the structure of the system. For instance, it can be rather complex, or the process of the failure development cannot be described in a simple way. In these case a "proper model" can help a lot in deriving reliability characteristics. The purpose of this paper is to introduce a generalized extended inverse Weibull software reliability model which includes both increasing/decreasing nature of the hazard function. The model can be effectively used for deriving and analyzing the corresponding failure rate. Reliability and other relevant measures are then computed from the fitted model.

The remainder of the paper is organized as follows: In Section 2, we describe a finite failure NHPP class of software reliability growth model (SRGM), and offer a decomposition of mean value function (MVF) to the finite failure NHPP models which enables us to relate the nature of failure intensity of the software to the hazard function and examine the suitability of some finite failure models. In Section 3, we discuss some of the existing finite failure reliability models and also we discuss the need for the development of new model which takes into account both increasing/decreasing nature of hazard function. In Section 4, we propose a generalized extended inverse Weibull finite failure NHPP model which describes the increasing/decreasing nature of the failure occurrence rate per fault and we also discuss the estimation of unknown model parameters by maximum method. The MLE's are consistent and asymptotically normally distributed as the sample size increases; see Zhao Xie (1996). In Section 5, we discuss the procedure of data analysis for software reliability assessment based on three different data sets DS I, DS II and DS III. Finally, Section 6 contains the major conclusions of the study.

2. Software Reliability Growth Model (SRGM)

Non-homogeneous Poisson Process (NHPP) model: During the testing phase, computer software is subject to software failures caused by errors latent in the software [Yamada

(1991)]. Test data such as times of software failures or the numbers of detected errors can be observed. If it is assumed that the correction of errors does not introduce any new errors, the cumulative number of detected errors increases as they are corrected and the mean time interval between software errors becomes longer. This means that the probability of no failure occurring in a fixed time interval, that is, the reliability increases with the progress of software testing. A mathematical tool that describes such an error-detection or failure occurrence phenomenon is called a software reliability growth model (SRGM). Many SRGM's have been developed for measuring an assessing software reliability, for details readers may refer to Jelinski and Moranda (1972), Goel and Okumoto (1979), and Musa (1980).

This is a class of time-domain data where software failures display the behavior of a non-homogeneous Poisson process [see Obha and Yamada (1984)]. The non-homogeneous Poisson process model (NHPP) that represents the number of failures experienced up to time t is a non-homogeneous Poisson process $\{N(t),\ t \geq 0\}$. The main issue in the NHPP model is to determine an appropriate mean value function to denote the expected number of failures experienced up to a certain time. Consider a non-homogeneous Poisson process with intensity $\lambda(t)$ at time t. The parameter $\lambda(t)$ denotes the failure intensity of the software at time t, is time dependent. Let $N(t)$ denote the cumulative number of faults detected by time t, and $m(t)$ denotes its expectation. Then $m(t) = E[N(t)]$, and the failure intensity $\lambda(t)$ is related as follows:

$$m(t) = \int_0^t \lambda\,(s)ds \quad \text{and} \quad \frac{\partial m(t)}{\partial t} = \lambda(t)$$

The NHPP model is based on the following assumptions:

1. The failure has an independent increment i.e. the number of failure during the time interval $(t, t+s)$ depends on the current time t and the length of the time interval s, and does not depend on the past history of the process.

2. The failure rate of the process is given by:
   P{exactly one failure in $(t, t+\Delta t)$ } $= P\{N(t+\Delta t) - N(t) = 1\} = \lambda(t)\Delta t + o(\Delta t)$
   where $\lambda(t)$ is the intensity function.

3. During a small interval Δt, the probability of exactly one failure is negligible, that is, P{two or more failure in $(t, t+\Delta t)$}$= o(\Delta t)$.

4. The initial condition is $N(0) = 0$.

On the basis of these assumptions the probability of exactly n failures occurring during the time interval $(0, t)$ for the NHPP is given by:

$$P\{N(t) = n\} = \frac{[m(t)]^n}{n!} e^{-m(t)} \quad , \quad n = 0,1,2, \dots, \infty.$$

Various time domain models have been proposed in literature which describes the stochastic failure process by a NHPP. These models differ in their failure intensity function $\lambda(t)$ and hence $m(t)$. The NHPP models can further be classified into finite failure and infinite failure categories. Finite failure NHPP models assume that the expected number of faults detected given infinite amount of time will be finite, whereas infinite failure models assume that an infinite number of faults that would be detected in infinite testing time.

Let a denote the expected number of faults that would be given infinite testing time in case of finite failure NHPP models. Then, the MVF of the finite failure NHPP models can also be written as:

$$m(t) = aF(t) \tag{2.1}$$

where $F(t)$ is the distribution function and $a = m(\infty)$. From the equation (2.1), the instantaneous failure intensity, $\lambda(t)$ for the finite failure NHPP model is given by: $\lambda(t) = aF'(t)$ and this can be written as:

$$\lambda(t) = [a - m(t)]\frac{F'(t)}{1-F(t)} = [a - m(t)]h(t) \tag{2.2}$$

where $h(t)$ is the hazard function or the failure occurrence rate per fault of the software, or the rate at which the individual faults manifest themselves as failures during testing. The quantity $[a - m(t)]$ denotes the expected number of faults remaining in the software at time t. Since $[a - m(t)]$ is a monotonically non-increasing function of time, the nature of overall failure intensity $\lambda(t)$, is governed by the nature of failure occurrence rate per fault $h(t)$, from the equation (2.2).

3. Some Existing NHPP Models

A number of analytical models have been proposed to address the problem of software reliability measurements. These approaches are based mainly on the failure history of software and can be classified according to nature of the failure process studied as indicated below:

1. Times between failure models
2. Failure count models
3. Fault seeding models
4. Input domain based model

[Jelinski and Moranda (JM) (1972)] model comes under the class times between failure. This is one of the earliest and probably the most commonly used model for assessing software reliability. A lot of variations in the JM model are proposed to describe testing situations where faults are not removed until the occurrence of a fatal one at which time the accumulated group of faults is removed. Schick and Wolverton (1973) model is based on the same assumption as the JM model except that the hazard function is assumed to be proportional to the current fault content of the program as well as to the failure time elapsed since the last failure.

The above two models assume that the faults are removed with certainty when detected. However, in practice Thayer et al. (1976) showed that it is not always the case. To overcome this limitation, Goel and Okumoto (1979b) proposed an imperfect debugging model which is basically an extension of the JM model. Littlewood (1980) took a different approach to the development of a model for times between failures. He argued that software reliability should not be specified in terms of the number of errors in the program.

Most popular and most basic fault seeding model is Mill's (1972) hypergeometric model. This model requires that the number of faults be randomly seeded in the program to be tested. Nelson (1978) model is an input domain based model, the reliability of a software is measured by running the software for a sample of n inputs. The n inputs are randomly chosen from the input domain set.

Langberg and Singapurwalla (LS) (1985) have developed a model by introducing total number of program instructions or commands into the failure rate. Van Pul (1992) has

considred a general class of failure rate intensity functions. A unification of SRM's is proposed by Chen and Sigapurwalla (1997) by self-exciting point process technique and they showed some of the existing models are special case of this model. Zhang et al. (2007) had introduced concept of equivalent failure time into JM model and its applications. Recently Mahapatra and Roy (2012) have developed a modified JM model which assumes the imperfect debugging process in fault removal activity during the testing phase. In this model authors assume that whenever a failure occurs, the detected fault is not perfectly removed and there is a chance of introducing new fault/s due to wrong diagnosis or incorrect modification in the software.

The failure occurrence rate per fault $h(t)$ can be a constant, increasing, decreasing, or increasing/decreasing. Here we describe some of the existing finite failure NHPP models, along with their hazard functions. The Goel-Okumoto (GO) model has received a lot of attention in the literature in software reliability modeling. Where a is the expected initial fault content prior to the testing, and b is the failure detection rate per fault. The failure occurrence rate per fault is constant in the case of the GO model. Musa-Okumoto model [Musa and Okumoto (1985)] is similar to the GO model, the primary difference being that it is based on execution time data, whereas the GO model uses the calendar time failure data. Table 1 gives the expressions for $m(t), \lambda(t)$ and $h(t)$ for the GO model.

However, in most of the real life testing scenario, the software failure intensity increases initially and then decrease. The generalized GO model [Goel (1985)] had captured this notion of increasing/decreasing nature of the failure intensity the nature of the failure occurrence rate per fault is determined by the parameter γ, and is increasing for $\gamma > 1$ and decreasing for $\gamma < 1$. Table 1 gives expressions of $m(t), \lambda(t)$ and $h(t)$ for the Goel-Okumoto, Weibull and delayed S-shaped models. The NHPP delayed S-shaped model is a stochastic model for a software error detection process based on NHPP in which the growth curve of the number of detected software errors for the observed failure data is S-shaped, called delayed S-shaped NHPP model [Yamada et al (1984)]. The software reliability growth curve is an S-shaped curve which means that the curve crosses the exponential curve from below and the crossing occurs once and only once. The detection rate of faults, where error detection rate changes with time, becomes the greatest at a certain time after testing begins after which it decreases exponentially. The S-shaped SRGM captures the software error removal phenomenon in which there is a time dela between the actual detection of the fault and its reporting. The testing process in this case can be seen as consisting of two phases: fault detection and fault isolation. The expressions for $m(t), \lambda(t)$ and $h(t)$ for the S-shaped model are presented in Table 1.

Table 1: Existing Finite Failure NHPP Models

Model Name	$m(t)$	$\lambda(t)$	$h(t)$
GO	$a\left(1-e^{-bt}\right)$	abe^{-bt}	b
Weibull	$a\left(1-e^{-bt^{\gamma}}\right)$	$ab\gamma t^{\gamma-1}e^{-bt^{\gamma}}$	$b\gamma t^{\gamma-1}$
S-shaped	$a[1-(1+bt)e^{-bt}]$	ab^2te^{-bt}	$\frac{b^2t}{1+bt}$

We present graph of data set DS I only as explained in Section 6 which led us to the development of new SRMs. The data set consists of 26 failures in 250 days (DS I). The plot of hazard rate of GO models, Weibull model and S-shaped model are shown in Figure 1. The plot of hazard rates of GO models, Weibull model and S-shaped model is plotted using the estimated values of parameters given in Table 3 of the respective models that is GO, Weibull and S-shaped models.

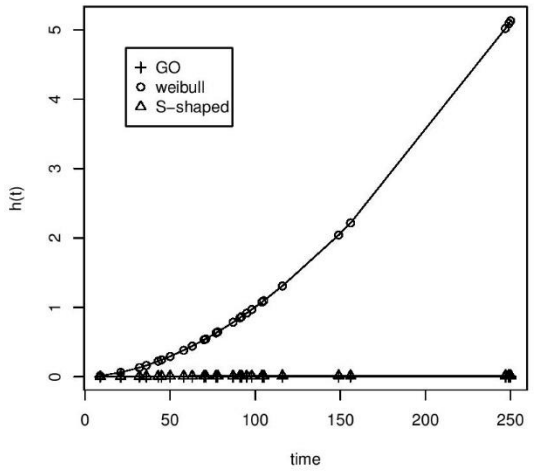

Figure 1: Plot of Hazard Rate for Existing NHPP Models

In reality this may not true because in the early stage of testing the tester is new to the software environment and they need time to adjust. This implies, less detection of failure caused by faults at the beginning. As testing progresses testers get good exposure to the software. This implies that there is an increase in the detection of failures and thus detection of faults increases in this period of time. When most of the software faults are eliminated, failures of software decreases as time increases. This idea leads us to the developed of a new model taking into account both increasing/decreasing behavior of the software failures by its hazard function.

4. Generalized Extended Inverse Weibull Finite Failure NHPP Model

It is well known that the Weibull distribution is one of the most widely used lifetime distribution in reliability engineering. The Weibull distribution has been used quite extensively when the data indicate a monotone hazard function (refer to Figure 1), i.e., the Weibull distribution does not exhibit a bathtub or upside-down bathtub shaped hazard rate function. Due to this limitation it cannot be used at all if the data indicate a non-monotone and unimodal hazard function. Thus it cannot be used to model the complex life time of a system. Hence, a number of extensions of the Weibull distribution are introduced to overcome this shortage. For example, Mudholkar and Shrivastava (1993) proposed an exponential Weibull distribution. Xie and Lai (1995) presented an additive Weibull distribution. Marshal and Olkin (1997) studied a Marshal-Olkin extended Weibull distribution. Jiang and Murthy (1998) discussed the shape of the hazard rate function for the mixture Weibull distribution. Xie et al. (2002) introduced different modified Weibull distributions with bathtub shape failure rate function. Pham and Lai (2007) presented the elaborate overviews of the various developments in extensions of the Weibull distribution. More recently Xiuyun and Zaizai (2014) presented an extended Weibull distribution. In this

paper, we introduce a generalized extended inverse Weibull (GEIW) distribution which has an increasing/decreasing form of hazard function. If empirical studies indicate that the hazard function might be unimodal then the GEIW distribution may be appropriate model. Let $G(x)$ be the CDF of extended Weibull distribution discussed by Xiuyun and Zaizai (2014), is given by:

$$G(x) = 1 - \exp(-\alpha x^{\beta} \exp\left(-\frac{\lambda}{x}\right)), \quad x > 0 \tag{4.1}$$

Let $t = \frac{1}{x}$ then, the CDF of extended inverse Weibull distributions can be defined by

$$G_1(t) = \exp\left(-\alpha t^{-\beta} \exp(-\lambda t)\right), \quad t > 0. \tag{4.2}$$

The probability density function of extended inverse Weibull distribution is given by:

$$g_1(t) = \alpha t^{-\beta}(\beta t^{-1} + \lambda) \exp\left(-\alpha t^{-\beta} \exp(-\lambda t)\right), \quad t > 0 \tag{4.3}$$

and zero otherwise. Where $\alpha > 0$ is scale parameter and $\beta > 0$ and $\lambda \geq 0$ are shape parameters.

The CDF of the generalized extended inverse Weibull distribution can be defined by:

$$F(t) = 1 - (1 - \exp\left(-\alpha t^{-\beta} \exp(-\lambda t)\right)^{\theta}, \ t > 0 \tag{4.4}$$

The probability density function of generalized extended inverse Weibull distribution is given by:

$$f(t) = \theta\alpha t^{-\beta}(\beta t^{-1} + \lambda)\exp(-\lambda t - \alpha t^{-\beta} \exp(-\lambda t) \left(1 - \exp\left(-\alpha t^{-\beta} \exp(-\lambda t)\right)\right)^{\theta-1} \tag{4.5}$$

and zero otherwise. Where $\alpha > 0$ is scale parameter and $\beta > 0$, $\lambda \geq 0$ and $\theta > 0$ are shape parameters.

We observe that this model has simple explicit formula and it does not involve any special functions. In this section, we develop a SRM which includes both increasing/decreasing behavior of the failure occurrence rate per fault can be taken into account by the hazard of GEIW model. The hazard rate function $h(t)$ of the generalized extended inverse Weibull SRM is given as follows:

$$h(t) = \theta\alpha\, t^{-\beta}(\beta t^{-1} + \lambda) exp\left(-\lambda t - \alpha t^{-\beta} \exp(-\lambda t)\right) \left[1 - \exp\left(-\alpha t^{-\beta} \exp(-\lambda t)\right)\right]^{-1} \tag{4.6}$$

The parameters $\lambda \geq 0$ and $\beta > 0$ are the shape parameters. The shape parameters govern the shape of the pdf, the hazard function and the general properties of the extended inverse Weibull distribution. The hazard function takes the increasing/decreasing nature of failure of software system into account. The corresponding MVF $m(t)$ is

$$\begin{aligned} m(t) &= aF(t) \\ &= a\left(1 - \left(1 - \exp\left(-\alpha t^{-\beta} \exp(-\lambda t)\right)\right)^{\theta}\right) \end{aligned} \tag{4.7}$$

it is noted that: $m(0) = 0$ and $m(\infty) = a$ and failure intensity function is

$$\lambda(t) = a\alpha t^{-\beta}(\beta t^{-1} + \lambda)\exp(-\lambda t - \alpha t^{-\beta} \exp(-\lambda t)) \quad \theta\left(1 - \exp\left(-\alpha t^{-\beta} \exp(-\lambda t)\right)\right)^{\theta-1} \tag{4.8}$$

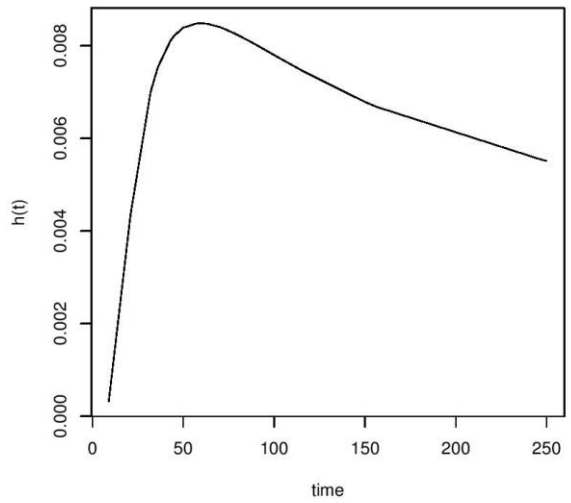

Figure 2: Hazard Rates of Generalized Extended Inverse Weibull (GEIW) Model for the data set DS-I

In Figure 2, we plot the hazard rates of our proposed GEIW model for that data set DS I only using the estimated values of parameters given in Tables 3. From the Figure 2, we observe that the estimated hazard rate (or failure occurrence rate) of generalized extended inverse Weibull (GEIW) Model is increasing in the beginning of software failure time till 50 days and decreasing linearly after failure time 50 (days).

Software Reliability

Let S_i $(i = 1,2,3, \dots)$ be a random variable representing the i^{th} software failure occurrence time. Define $X_i = S_i - S_{i-1}, (i = 1,2,3, \dots); S_0 = 0$ is a random variable representing the time interval between the $(i-1)^{th}$ and i^{th} software failure occurrence. The conditional probability that the i^{th} software failures does not occur between $(t, t+x], (x \geq 0)$ on the condition that the $(i-1)^{th}$ software failure has occurred at testing time t, is given by:

$$R(x|t) = P\{X_i > x | S_{i-1} = t\} = \exp[-\{m(x+t) - m(t)\}], \; t \geq 0, x \geq 0 \qquad (4.9)$$

Substituting $m(t)$ in the equation (4.9), we have the software reliability for finite failure generalized extended inverse Weibull model as:

$$R(x|t) = \exp\left[-\left\{a\left(1-\left(1-exp\left(-\alpha(t+x)^{-\beta}\, exp\left(-\lambda(t+x)\right)\right)\right)^{\theta}\right) - a\left(1-\left(1-exp\left(-\alpha t^{-\beta}\, exp(-\lambda t)\right)\right)^{\theta}\right)\right\}\right], \; t \geq 0, x \geq 0 \qquad (4.10)$$

The reliability function $R(t)$, defined as the probability that there are no failures in the time interval $(0, t)$ is given by $R(t) = P[N(t) = 0] = e^{-m(t)}$.

5.Parameters Estimation

Parameter estimation is of primary importance in software reliability data analysis. We discuss statistical inference procedure for the NHPP model discussed in Section 4 based on a method of maximum likelihood which is the most important and widely used formal

estimation technique. We now proceed for estimating parameters of generalized extended inverse Weibull software reliability model using interval domain data.

Estimation using interval Domain Data

Suppose that n pairs of observations $(t_i, y_i)(i = 1,2,3, \dots, n; 0 < t_1 < t_2 < \cdots < t_n)$ are made during the testing phase where y_i denote the number of software faults detected up to the testing time $t_i, i = 1,2, \dots, n$. The corresponding probability mass function is $P[N(t_1) = y_1, N(t_2) = y_2, \dots, N(t_n) = y_n]$ and hence the likelihood function for the interval domain data is given by

$$L = \prod_{i=1}^{n} \frac{(m(t_i) - m(t_{i-1}))^{y_i - y_{i-1}}}{(y_i - y_{i-1})!} \exp[-m(t_i) - m(t_{i-1})] \quad (5.1)$$

Where $t_0 = 0$ and $y_0 = 0$. Taking the natural logarithm of the equation (5.1) yields log-likelihood function, that is

$$lnL = \sum_{i=1}^{n} (y_i - y_{i-1}) \ln[m(t_i) - m(t_{i-1})] - m(t_n) - \sum_{i=1}^{n} \ln(y_i - y_{i-1})! \quad (5.2)$$

substituting mean value function $m(t)$ into the equation (5.2), we get logarithmic of likelihood function as (excluding the terms independent of parameters),

$$lnL = \sum_{i=1}^{n} (y_i - y_{i-1}) \ln\left[a\left(1 - \left(1 - \exp\left(-\alpha t_i^{-\beta} \exp(-\lambda t_i)\right)\right)^{\theta}\right) - a\left(1 - \left(1 - \exp\left(-\alpha t_{i-1}^{-\beta} \exp(-\lambda t_{i-1})\right)\right)^{\theta}\right)\right] - a\left(1 - \left(1 - \exp\left(-\alpha t_n^{-\beta} \exp(-\lambda t_n)\right)\right)^{\theta}\right) \quad (5.3)$$

Taking partial derivative of the equation (5.3) w.r.t. $a\ \alpha, \beta, \lambda, \theta$ and equating to zero we get the likelihood equation. Closed form expressions for MLEs of $a, \alpha, \beta, \lambda$ and θ cannot be obtained. However, the MLEs can be obtained by iterative solution procedure. Setting the derivatives of the log-likelihood functions for $(a, \alpha, \beta, \lambda, \theta)$ to zero, the MLEs $(\hat{a}, \hat{\alpha}, \hat{\beta}, \hat{\lambda}, \hat{\theta})$ are obtained by iterative solution procedure i.e., Newton-Raphson method. We have used R-software for the iterative solution procedure. Let $\hat{a}, \hat{\alpha}, \hat{\beta}, \hat{\lambda}$ and $\hat{\theta}$ be the MLEs of $a, \alpha, \beta, \lambda$ and θ respectively. Then MLE of MVF and intensity function is obtained by replacing $a, \alpha, \beta, \lambda$ and θ by its MLEs $\hat{a}, \hat{\alpha}, \hat{\beta}, \hat{\lambda}$ and $\hat{\theta}$ in the equations (4.7) and (4.8) respectively. Software reliability can be estimated from the equation (4.10).

6. Analysis of Three Data Sets

The Data Set I is about US Naval Tactical Data Systems (NTDS) given by [Goel and Okumoto (1979a)]: the software data set was extracted from information about failures in the development of software for the real time multi-computer complex of the US Naval Fleet Computer Programming Centre of the US Naval Tactical Data System (NTDS). The software consists of 38 different project modules. The time horizon is divided into four phases: Production phase, test phase, user phase and subsequent test phase. The 26 software failures were found during the production phase, five during the test phase and the last failure was found on 4th Jan 1971. One failure was observed during the user phase, in September 1971, and two failures during the test phase in 1971. The Data Set II is about Online Data Entry IBM Software Package: The data reported by [Obha (1984)] are recorded from testing an on-line data entry software package developed at IBM. The Data Set III is

about Real-Time Command and Control System: - The data set was reported by [Musa et al. (1987)] based on failure data from a real-time command and control system, which represents the failure during system testing for 25 hours of CPU time.

Goodness of Fit Test for the GEIW Model:

It is important to perform goodness-of –fit test to check statistically whether the applied software reliability model (SRGM) provides a good fit with the observed data. We discuss

Table 2: MLEs, K-S Statistics, p-values for GEIW Model

Data Set No.	K-S Statistic	p-value
DS-I	0.136865	0.66484
DS-II	0.15877	0.58158
DS-III	0.077251	0.99991

The goodness of fit test based on the Kolmogorov-Smirnov (K-S) test statistics [Conover (1980)] for an NHPP model [Yamada (1991)], which is useful even if the sample size of the observed data is small. The Kolmogorov-Smirnov test statistic is given by: $D = max(D_i)$,

where $i = 1,2,3, \dots, n - i$

$$D_i = \max\left\{\left|\frac{\widehat{m(t_i)}}{t_n} - \frac{i}{n-1}\right|, \left|\frac{\widehat{m(t_i)}}{t_n} - \frac{i-1}{n-1}\right|\right\} \quad (6.1)$$

For the failure-occurrence time data $t_i, (i = 1,2,3, \dots, n)$. Using the value of the test statistic in the equation (6.1), p-value is computed using the formula for the sample size n for the specified level of significance α. If the p-value is greater then, the specified level of significance α then we may conclude that the observed data fits the applied software reliability growth model (SRGM).

We used the Kolmogorov-Smirnov goodness-of-fit test for checking the adequacy of the model. For details of this test see Goel (1982). Basically, the test provides a statistical comparison between the actual data and the model proposed. This test is applied to all the three data sets DS-I, DS-II and DS-III, i.e. K-S statistics and p-values are computed from the values of K-S statistic and p-value in the Table 2, we conclude that the proposed GEIW model fits well for all the three data sets, so we conclude that the proposed model could be considered to be a good choice for the data.

Applying data sets DS I, DS II and DS III on the generalized extended inverse Weibull model discussed above, we obtain the maximum likelihood (ML) estimates and goodness of fit measures, such a sum of squares due to error (SSE), mean squares error (MSE), predictive ratio risk (PRR) [Pham and Deng (2003)] and Akaikes information criterion (AIC) for the model comparison. The values of SSE, MSE, PRR and AIC for existing and for the proposed finite failure models are summarized in Table 3, 4 and 5 respectively.

Further we have plotted the actual data and the estimated values of cumulative faults (or Mean Value Function (MVF)) of finite models against time for data sets DS I, DS II, DS III as shown in Figures 3, 4 and 5 respectively. The existing models like GO, Weibull and S-shaped models and the models finite generalized exponential distribution(FGED), finite generalized inverse exponential distribution(FGIED), proposed by Manjunatha and Harishchandra (2011), generalized inverse Weibull(GIW) and extended inverse Weibull(EIW) proposed by Hanagal and Bhalerao (2016, 2017) are compared with the proposed generalized extended inverse Weibull(GEIW) finite failure NHPP model.

Table 3: MLEs and AICs of Finite Failure for DS I

Model	Estimates	SSE	MSE	PRR	AIC
GO	$\hat{a}$ =34.0152 $\hat{b}$ =0.0058	129.0067	5.3753	1.4056	169.3803
Weibull	$\hat{a}$ =15.3157 $\hat{b}$ =0.0001 $\hat{\gamma}$ =2.7791	2556.511	111.1527	24083.41	199.4800
S-Shaped	$\hat{a}$ =27.5041 $\hat{b}$ =0.0186	1208.813	50.3672	27.2446	169.8360
FGED	$\hat{a}$ =15.3157 $\hat{b}$ =0.0001 $\hat{\alpha}$ =2.7791	96.5955	4.1998	94.0674	167.5664
FGIED	$\hat{a}$ =15.3157 $\hat{b}$ =0.0001 $\hat{\alpha}$ =2.7791	65.5712	4.8986	2.8509	168.6538
GIW	$\hat{a}$ =36.8148 $\hat{\alpha}$ =40.66 $\hat{b}$ =0.99 $\hat{\gamma}$ =2.10	65.3842	2.972	4.6387	70.6464
EIW	$\hat{a}$ =33.2683 $\hat{\alpha}$ =42.0 $\hat{\beta}$ =0.84 $\hat{\lambda}$ =0.002	58.0075	2.6367	0.5221	71.0302
GEIW	$\hat{a}$ =26.1 $\hat{\alpha}$ =52 $\hat{\beta}$ =0.002 $\hat{\lambda}$ =0.002 $\hat{\theta}$ =2	50.941261	2.42577	0.2507	68.7395

From the Table 3, we observe that the SSE, MSE, PRR and AIC values of our proposed generalized extended inverse Weibull (GEIW) model are less as compared to the existing models. Hence we conclude that our proposed model performs better than the existing models for this data set i.e., DS I.

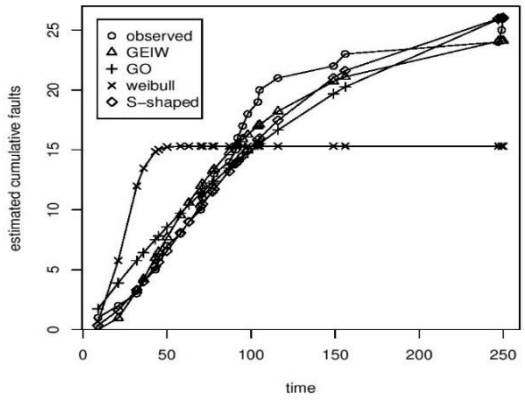

Figure 3: Plot of data and Estimated Cumulative Faults against time for Data set I

From the Figure 3, we observe that the shape of the mean value function for the proposed generalized extended inverse Weibull model is closer to the shape of mean value function of

the observed data, as compared to GO, Weibull and S-shaped models for data set DS I. Hence from Figure 3, we conclude that our proposed model GEIW explains the data better than GO, Weibull and S-shaped models.

From the Table 4, we observe that the SSE, MSE, PRR and AIC values of our proposed generalized extended inverse Weibull (GEIW) model are less as compared to the existing models. Hence we conclude that our proposed model performs better than the existing models for this data set i.e., DS II.

Table 4: MLEs and AICs of Finite Failures for DS II

Model	Estimates	SSE	MSE	PRR	AIC
GO	$\hat{a}$ =25.4892 $\hat{b}$ =0.0029	44.4372	2.2219	18.8628	195.3148
Weibull	$\hat{a}$ =29.3242 $\hat{b}$ =0.0058 $\hat{\gamma}$ =0.8400	39.8159	2.0956	4.6564	193.8581
S-Shaped	$\hat{a}$ =22.6199 $\hat{b}$ =0.0080	467.2066	23.2603	12460.87	202.2326
FGED	$\hat{a}$ =27.7642 $\hat{b}$ =0.0021 $\hat{\alpha}$ =0.8147	156.7386	8.2494	17.0705	193.9031
FGIED	$\hat{a}$ =91.9977 $\hat{b}$ =18.0629 $\hat{\alpha}$ =0.0720	40.0537	2.1081	4.4855	193.7165
GIW	$\hat{a}$ =24.9348 $\hat{\alpha}$ =117 $\hat{b}$ =1.05 $\hat{\gamma}$ =1.09	22.5784	1.2544	0.087	84.844
EIW	$\hat{a}$ =26.8684 $\hat{\alpha}$ =8.7 $\hat{\beta}$ =0.37 $\hat{\lambda}$ =0.002	37.345	2.0747	0.3613	67.66
GEIW	$\hat{a}$ =25 $\hat{\alpha}$ =6.2 $\hat{\beta}$ =0.31 $\hat{\lambda}$ =0.003 $\hat{\theta}$ =0.9	33.73487	1.984404	0.230296	65.4064

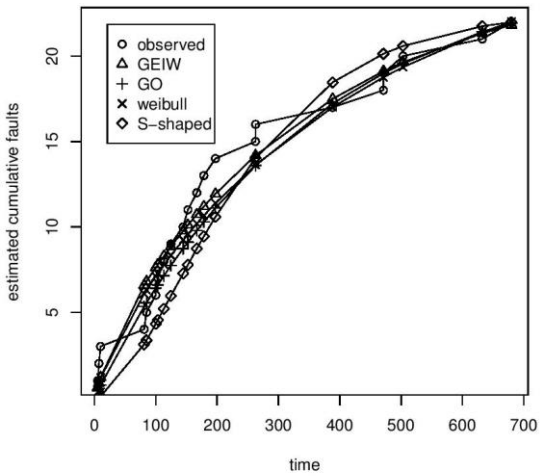

Figure 4: Plot od Data and Estimated Cumulative Faults against Time for DS II

Table 5: MLEs and AICs of Finite Failures for DS III

Model	Estimates	SSE	MSE	PRR	AIC
GO	$\hat{a}$ =23.4121 $\hat{b}$ =0.0035	5.8191	0.4476	0.4104	124.2018
Weibull	$\hat{a}$ =3.5128 $\hat{b}$ =0.0001 $\hat{\gamma}$ =1.9695	771.0485	64.254	5472.873	151.4680
S-Shaped	$\hat{a}$ =16.4356 $\hat{b}$ =0.0137	24.3006	1.8693	54.5298	124.4217
FGED	$\hat{a}$ =21.1765 $\hat{b}$ =0.0044 $\hat{\alpha}$ =1.1052	3.825	0.3188	11.6132	124.1545
FGIED	$\hat{a}$ =63.2471 $\hat{b}$ =36.6030 $\hat{\alpha}$ =0.1230	6.9954	0.8851	0.5829	123.8616
GIW	$\hat{a}$ =19.4955 $\hat{\alpha}$ =30 $\hat{b}$ =1.05 $\hat{\gamma}$ =2.9	17.1171	1.5561	104.2216	43.73
EIW	$\hat{a}$ =27.6785 $\hat{\alpha}$ =15.0 $\hat{\beta}$ =0.51 $\hat{\lambda}$ =0.001	2.7	0.2538	0.1380	37.58
GEIW	$\hat{a}$ =37 $\hat{\alpha}$ =9 $\hat{\beta}$ =0.39 $\hat{\lambda}$ =0.001 $\hat{\theta}$ =0.75	2.108723	0.210872	0.050359	35.3

From the Table 5, we observe that the SSE, MSE, PRR and AIC values of our proposed generalized extended inverse Weibull (GEIW) model are less as compared to the existing models. Hence we conclude that our proposed model performs better than the existing models for this data set i.e., DS III.

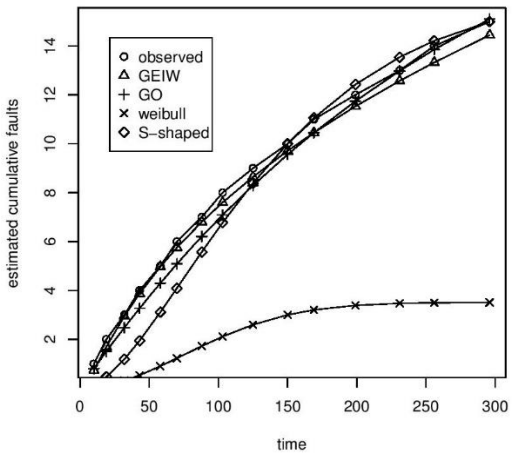

Figure 5: Plot of Data and Estimate Cumulative Faults against Time of DS III

From the Figure 5, we observe that the shape of the mean value function for the proposed generalized extended inverse Weibull model is closer to the shape of mean value function of the observed data, as compared to GO, Weibull and S-shaped models for data set DS III. Hence from Figure 5, we conclude that our proposed model GEIW explains the data better than GO, Weibull and S-shaped models

7. Conclusions and Remarks

Since the existing finite failure non-homogeneous Poisson process (NHPP) models are inadequate to describe failure process underlying increasing/decreasing phenomenon, in this paper we have proposed an alternative model namely the generalized extended inverse Weibull software reliability growth model. We use decomposition of the mean value function (MVF) of finite failure NHPP model to enable us to include the increasing and decreasing nature of the failure intensity to the failure occurrence rate per fault. The model parameters are estimated using ML method by using the interval domain data type. Kolmogorov-Smirnov goodness-of-fit test is carried out for the proposed model and the proposed model fits well for all the three data sets. The different existing models and the proposed model is compared using or sum of squares due to error, mean squares error, predictive ratio risk and Akaikes information criteria. It may be observed that the proposed model performs better than the existing finite failure category models.

Acknowledgments

We thank the referee for the valuable suggestions and comments which resulted in an improved version of the earlier manuscript.

References

1. Chen, Y. and Singapurwalla, N. D. (1997). Unification of software reliability models by self-exciting point processes. Advances Application Probability, 29, 337-352.

2. Chen, Z. (2010). Empirical Bays analysis on the power law process with natural conjugate priors. Journal of Data Science, 8(1), 151-172.

3. Conover, W. J. (1980). Practical Nonparametric Statistics (2nd ed.). John Wiley and Sons, New York.

4. Goel, A. L. and Okumoto, K. (1979a). A Markovian model for reliability and other performances measures of software system. Proceedings of COMPCON (IEEE Computer Society Press, Los Angeles), 769-774.

5. Goel, A. L. and Okumoto, K. (1979b). Time dependent error-detection rate model for software reliability and other performance measures. IEEE Transactions on Reliability, 28(3), 206-211.

6. Goel, A. L. (1985). Software reliability models: limitations and applicability, IEEE Transactions on Software Engineering, 2(12), 1411-1423.

7. Hanagal, D. D. and Bhalerao, N. N. (2016). Modeling and statistical inference on generalized inverse Weibull software reliability growth. Journal of the Indian Society for Probability and Statistics, 17(2), 161-184.

8. Hanagal, D. D. and Bhalerao, N. N. (2017). Modeling on extended inverse Weibull software reliability growth model. Indian Association for Product, Quality, and Reliability Transactions, 42(2), 75-95.

9. Hanagal, D. D. and Bhalerao, N. N. (2018). Analysis of delayed S-shaped software reliability growth model with time dependent fault content rate function. Journal of Data Science, 16(4), 857-878.

10. Hjorth, U. (1980). A reliability distribution with increasing, decreasing, constant and bathtub-shaped failure rate. Technometrics, 22, 99-107

11. Jelinski, Z. Moranda, P. B. (1972). Software Reliability Research in Statistical Computer Performance Evaluation. W. Freiberger (ed), Academic Press, New York, 465-297.

12. Jiang, R. and Murthy, D. N. P. (1998). Mixture of Weibull distributions parametric characteristics of failure rate function. Applied Stochastic Models and Data Analysis, 14, 47-65.

13. Langberg, N. and Singapurwalla, N. D. (1985). Unification of some software reliability models via the Bayesian approach. SIAM J. Sci. Statist. Comput., 6, 781-790.

14. Littlewood, B. (1980). Theories of software reliability: How good are they and how can they be improved? IEEE Trans. Software Engg. 6, 489-500.

15. Lyu, M. R. (1996). Handbook of Software Reliability Engineering, McGraw-Hill, New York.

16. Mahapatra, G. S. and Roy, P. (2012). Modified Jelinski-Moranda software reliability model with imperfect debugging phenomenon. International Journal of Computer Applications, 48, 38-46.

17. Manjunatha, K. M. and Harishchandra, K. (2011). Modeling and statistical inference on generalized inverse exponential software reliability growth model. Far East Journal of Theoretical Statistics, 39(1), 67-77.

18. Marshall, A. W. and Olkin, I. (1997). A new method for adding a parameter to a family of distributions with applications to the exponential and Weibull families. Biometrika, 84, 641-652.

19. Mills, H. D. (1972). On the statistical validation of computer programs. IBM Federal Syst. Div., Gaithersburg, MD, Rep. 72-6015

20. Mudholkar, G. S. and Shrivastava, D. K. (1993). Exponential Weibull family for analyzing bathtub failure rare data. IEEE Transactions on Reliability, 42(2), 299-302.

21. Musa, J. D. (1980). The measurement and management of software reliability. Proc.

IEEE, 68(9), 1131-1143.

22. Musa, J. D. and Okumoto, K. (1985). Applications of basic and logarithmic Poisson execution model in software reliability measure. The Challenges of Advanced Computing Technology to System Designs Methods, NATO Advanced Study Institute, 34(1), 68-100.

23. Musa, J. D., Lannio, A. and Okumoto, K. (1987). Software Reliability: Measurement, Prediction and Application, McGraw-Hill, New York.

24. Nelson, E. (1978). Estimating software reliability from test data, Micro-Electron, 17, 67-74.

25. Obha, M. and Yamada, S. (1984). S-shaped software reliability growth models. Proc. of 4th International Conference Reliability and Maintainability, 430-436.

26. Okumoto, K. and Goel, A. L. (1978). Availability and other performance measures of software systems under imperfect maintenance in Proc. COMPSAC, Chicago II., 66-71.
27. Pham, H. (2006). System Software Reliability, Springer, New York.

28. Pham, H. and Deng, C. (2003). Predictive-ratio risk criterion for selecting software reliability models. Proc. 9th International Conference on Reliability and Quality in Design, August. Honolulu, Hawaii, 17-21.

29. Pham, H. and Lai, C. D. (2007). On recent generalizations of Weibull distribution. IEEE Transactions on Reliability, 56, 454-462.

30. Schick, G. J. and Wolverton, R. W. (1973). Assessment of Software Reliability, Proceedings in Operations Research, Wirzberg-Wien: Physica Verlag, 365-422.

31. Singapurwalla, N. D. and Wilson, S. W. (1999). Statistical Methods in Software Engineering: Reliability and Risk, Springer-Verlag, New York.

32. Van Pul, M. C. (1992). Asymptotic properties of a class of statistical models in software reliability. Scandinavian Journal of Statistics, 19(3), 235-253.

33. Wood, A. (1996). Predicting software reliability. IEEE Computer, 11, 69-77.

34. Xie, M. and Lai, C. D. (1995). Reliability analysis using an additive Weibull model with bathtub-shaped failure rate function. Reliability Engineering and System Safety, 52, 87-93.

35. Xie, M., Tang, Y. and Goh, T. N. (2002). A modified Weibull Extension with bathtub-shaped failure rate function. Reliability Engineering and System Safety, 76, 279-285.

36. Xiuyun, P. and Zaizai, Y. (2014). Estimation and application for a new extended Weibull distribution. Reliability Engineering and System Safety, 121, 34-42.

37. Yamada, S. (1991). Software quality/reliability measurement and assessment: software reliability growth models and data analysis. Journal of Information Processing, 14(3), 254-266.

38. Yamada, S., Narihisa, H. and Osaki, S. (1984). Optimum release policies for a software system with a scheduled software delivery time. International Journal Systems Science, 15(8), 905-914.
39. Zhang, Y., Chen, W. and Sun, S. (2007). Improvement of the software reliability model with equivalent failure times. Proceedings of the 2007 WSEAS International Conference on Computer Engineering and Applications, Gold Coast, Australia, January 17-19, 84-88.

40. Zhao, M. and Xie, M. (1996). On maximum likelihood estimation for a general non-homogeneous Poisson process. Scandinavian J. Statistics, 23(4), 597-607.